

FinXpert

GitHub Link:

<https://github.com/RahulMahato23/FinXpert>



Presented by: Tech Titans

FinXpert

Home

Features

Technical Achievements

Team

Contact

Sign In / Register

Guest Login

The background of the page features a close-up photograph of several gold-colored coins scattered on a surface. A magnifying glass is held over the coins, with its lens focused on one particular coin in the center, symbolizing analysis or inspection.

EMPOWER YOUR FINANCIAL FUTURE WITH FINXPERT

Real-time Financial Insights.

GET STARTED

GUEST LOGIN

Wanna Try? Just login as guest and start using our app.

INDEX

- 1. Project Overview**
- 2. Need for the software**
- 3. Potential Users**
- 4. Major Requirements**
- 5. Design and implementation**
- 6. Future Scope**



PROJECT OVERVIEW



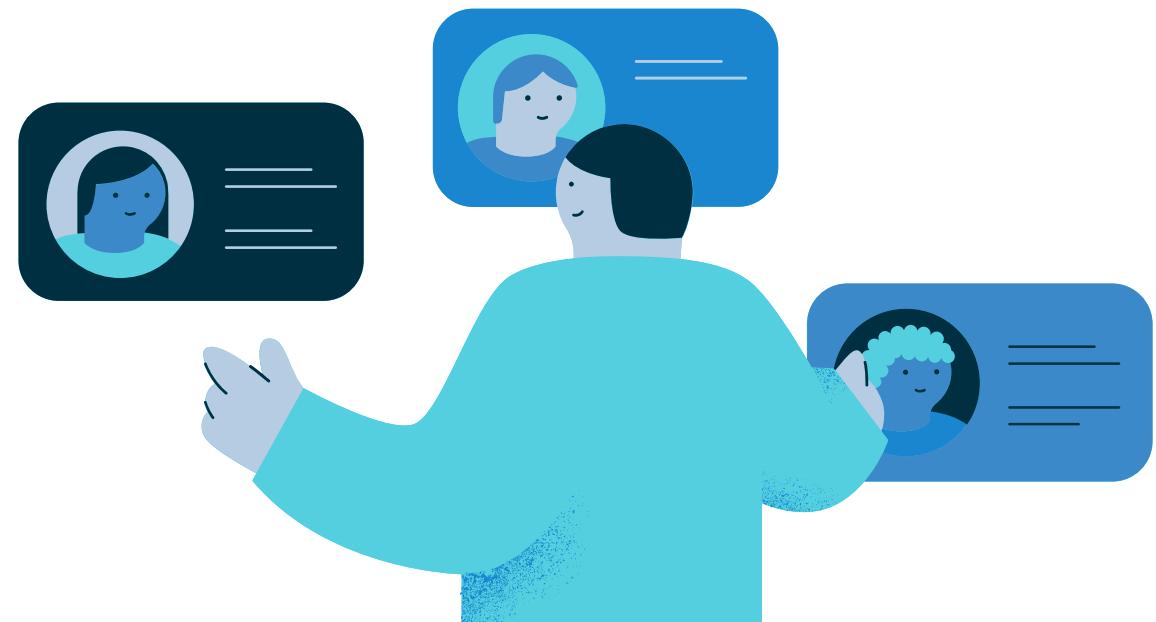
- **Tracks All Your Money**
- **Helps You Set Financial Goals**
- **Shows Your Money Mood**
- **Gives You Smart Advice**



FinXpert is like your personal money assistant. It helps you track your expenses and income, gives you smart suggestions based on your spending habits, and even lets you chat with an AI to understand your finances better.

NEED FOR THE SOFTWARE

- People need smarter ways to manage complex finances.
- Existing apps don't give personalized financial insights.
- AI can help users understand and improve spending habits.
- Users want everything—from tracking to analysis—in one place.
- Secure and simple access makes financial tools more usable.



POTENTIAL USERS

- Working professionals managing budgets.
- Students tracking expenses.
- Families planning household finances.
- Finance enthusiasts seeking insights.



MAJOR REQUIREMENTS

1.

Dashboard

- Show current month's total income and expenses
- Display expenditure-to-income ratio with actionable financial advice.
- Include interactive charts summarizing financial health and trends.



2.

Expense & Income Management

- Add, edit, delete, and view income and expense entries.
- Categorize transactions (e.g., groceries, travel, freelance).
- Attach cognitive triggers (e.g., impulse, social pressure) and moods for behavioral analysis.
- Filter transactions by date, category, trigger, mood, etc.

3.

Smart AI Assistant

- Allow users to interact using natural language (NLP).
- Enable conversational expense/income addition.
- Offer AI-powered financial summaries and personalized recommendations.
- Respond to queries like "How much did I spend on food last week?"



MAJOR REQUIREMENTS

4.

Financial Summary & Analysis

- Provide income vs expense breakdown.
- Analyze mood and cognitive triggers in spending.
- Show category-wise spending trends and historical analysis.
- AI-based suggestions to optimize savings and manage spending behavior.



5.

Ledger Views

- Tabular display of all income and expense entries.
- Support filtering, sorting, and data export.
- Ability to view full transaction details: title, amount, date, category, payment mode, mood, etc.

6.

Analytics & Reports

- View monthly summaries and trend visualizations.
- Generate category-based charts, pie graphs, and bar plots.
- Insights into spending spikes, seasonal patterns, and income sources.



MAJOR REQUIREMENTS



7.

User Authentication & Security

- OTP-based login system with rate limiting.
- JWT-based session management.
- Store sensitive data securely using Kubernetes Secrets and encrypted storage.

8.

Frontend Experience

- Built with React + Vite for a fast, responsive UI.
- Dark/light mode, animations (Framer Motion), and mobile optimization (in progress).

9.

Backend Architecture

- Microservices-based structure using Node.js.
- Core services: User, Expense, Income, Dashboard, AI, Orchestrator, Event Bus.
- Event-driven communication via Kafka.
- Databases: PostgreSQL (users), MongoDB (financial data).

10.

Testing & Validation

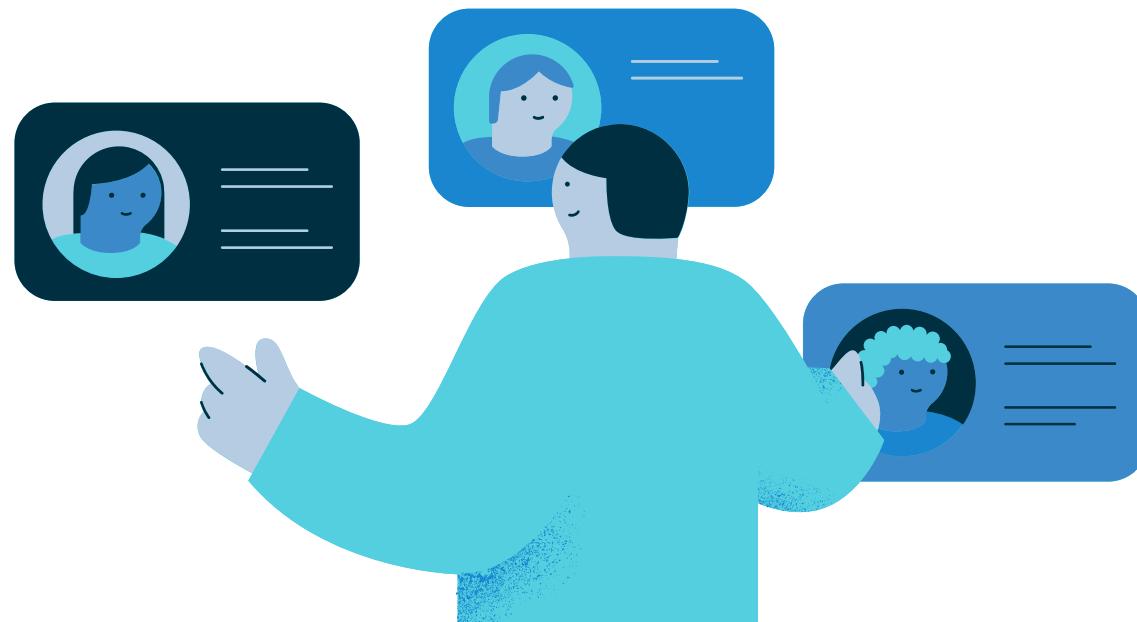
- Unit and integration testing for:
 - OTP, Expense/Income Add/Delete
 - Data queries and edge cases
- Manual and peer-reviewed testing approach.



DESIGN AND IMPLEMENTATION

DESIGN

- Microservices Architecture – Modular and scalable system using Docker for containerization and deployment
- Event-Driven Flow – Kafka for service communication
- Smart AI Module – Uses Open AI for insight generation
- Secure APIs – JWT-based protected routes
- Dynamic UI – React with page-wise routing



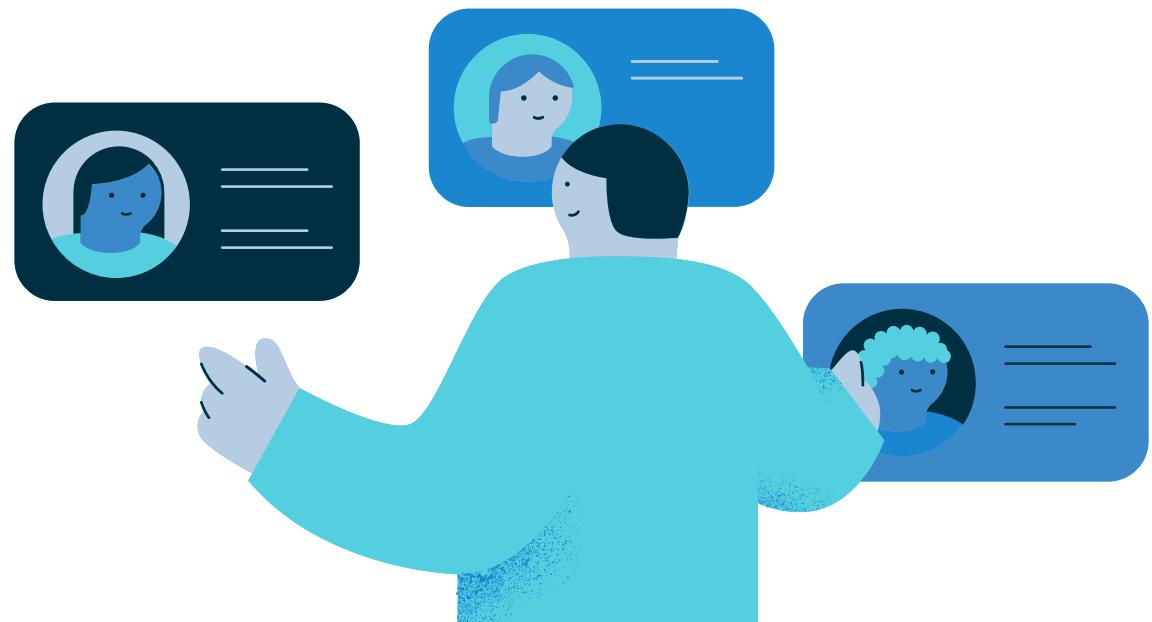
DESIGN AND IMPLEMENTATION

MERN

- M – MongoDB (NoSQL data storage)
- E – ExpressJS (API layer)
- R – ReactJS (Frontend)
- N – NodeJS (Service logic)

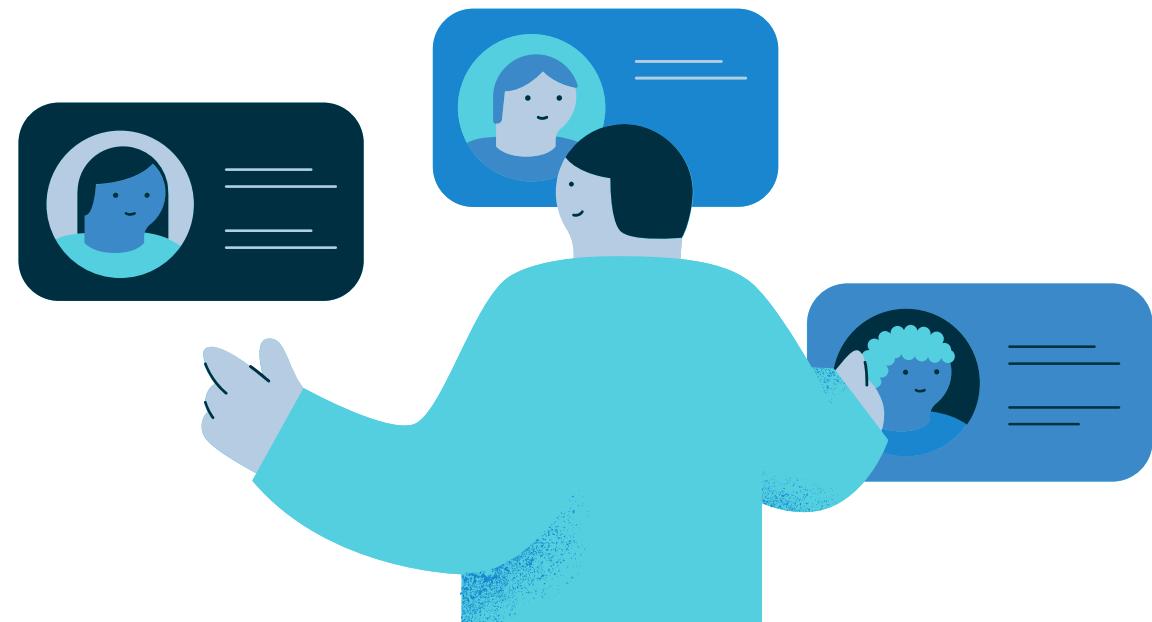
Add Ons:

- Kafka – Asynchronous communication between services
- OpenAI API – AI-based analysis & insights
- PostgreSQL – Structured storage for financial summaries & stats
- Twilio API – OTP-based user verification



FUTURE SCOPE

- Advanced AI recommendations.
- Budgeting and goal planning tools.
- Mobile app version.
- Notification and reminder system.
- ML-driven savings and investment suggestions.



- 
- A blurred background image of a construction site featuring a large yellow excavator and other heavy machinery under a clear blue sky.
- Coordination
 - Technical Skills
 - Team Spirit
 - Plan-driven Process

OUR MEMBERS



Dwij Om Osho



Sumit Kumar



Karthik S Pillai



Madhav Khetan



Aniket



Rahul Mahato



Archit Atrey



Devank



Ayush



Piyush



THANK YOU

